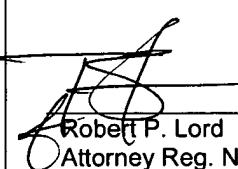
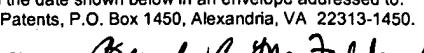




01-17-06

AFK  
17

TRANSMITTAL OF APPEAL BRIEF		Docket No. 16159/021001; P6416	
In re Application of: Syed M. Ali et al.			
Application No. 09/997,927-Conf. #5348	Filing Date November 30, 2001	Examiner D. M. Doan	Group Art Unit 2143
Invention: TRANSPARENT INJECTION OF INTELLIGENT PROXIES INTO EXISTING DISTRIBUTED APPLICATIONS			
<b><u>TO THE COMMISSIONER OF PATENTS:</u></b>			
Transmitted herewith is the Appeal Brief in this application, with respect to the Notice of Appeal filed: <u>November 14, 2005</u> .			
The fee for filing this Appeal Brief is <u>\$ 500.00</u> .			
<input checked="" type="checkbox"/> Large Entity		<input type="checkbox"/> Small Entity	
<input type="checkbox"/> A petition for extension of time is also enclosed.			
The fee for the extension of time is _____.			
<input type="checkbox"/> A check in the amount of _____ is enclosed.			
<input type="checkbox"/> Charge the amount of the fee to Deposit Account No. <u>50-0591</u> . This sheet is submitted in duplicate.			
<input checked="" type="checkbox"/> Payment by credit card. Form PTO-2038 is attached.			
<input checked="" type="checkbox"/> The Director is hereby authorized to charge any additional fees that may be required or credit any overpayment to Deposit Account No. <u>50-0591</u> . This sheet is submitted in duplicate.			
 Robert P. Lord Attorney Reg. No. : 46,479 OSHA · LIANG LLP 1221 McKinney St., Suite 2800 Houston, Texas 77010 (713) 228-8600		Dated: <u>January 17, 2006</u>	
<p>I hereby certify that this paper (along with any paper referred to as being attached or enclosed) is being deposited with the U.S. Postal Service as Express Mail, Airbill No. EV766455942US, on the date shown below in an envelope addressed to: MS Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.</p> <p>Dated: January 17, 2006      Signature:  (Brenda C. McFadden)</p>			



I hereby certify that this correspondence is being deposited with the U.S. Postal Service as Express Mail, Airbill No. EV766455942US, in an envelope addressed to: MS Appeal Brief – Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on the date shown below.

Dated: January 17, 2006

Signature:

*Brenda C. McFadden*  
(Brenda C. McFadden)

Docket No.: 16159/021001; P6416  
(PATENT)

## IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Syed M. Ali et al.

Conf. No.: 5348

Application No.: 09/997,927

Art Unit: 2143

Filed: November 30, 2001

Examiner: D. M. Doan

For: TRANSPARENT INJECTION OF  
INTELLIGENT PROXIES INTO EXISTING  
DISTRIBUTED APPLICATIONS

Assistant Commissioner for Patent  
Mail Stop Appeal Brief-Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

### APPELLANT'S BRIEF UNDER 37 C.F.R. § 41.37

Dear Sir:

Pursuant to 37 C.F.R. § 41.37, please consider the following Appellant's Brief in the referenced Application currently before the Board of Patent Appeals and Interferences.

01/19/2006 BABRAHAI 00000038 09997927

01 FC:1402

500.00 0P

## TABLE OF CONTENTS

I. Real Party in Interest .....	4
II. Related Appeals and Interferences .....	4
III. Status of Claims .....	4
IV. Status of Amendments .....	5
V. Summary of Claimed Subject Matter .....	5
VI. Grounds of Rejection to be Reviewed on Appeal .....	7
VII. Argument .....	7
A. Claims 2-8, 11-14, 16-24, and 32 .....	8
B. Claim 31 .....	11
C. Claim 10 .....	12
D. Claim 15 .....	13
VIII. Conclusion .....	14
Claims Appendix .....	15
Related Proceedings Appendix .....	19

**TABLE OF AUTHORITIES****CASES**

<i>Brown v. 3M</i> , 265 F.3d 1349 (Fed. Cir. 2001).....	7, 11, 12, 13, 14
<i>Diversitech Corp. v. Century Steps, Inc.</i> , 850 F.2d 675 (Fed. Cir. 1988).....	8
<i>Orthokinetics, Inc. v. Safety Travel Chairs, Inc.</i> , 806 F.2d 1565 (Fed. Cir. 1986).....	8
<i>Schering Corp. v. Geneva Pharms.</i> , 339 F.3d 1373 (Fed. Cir. 2003).....	7

**STATUTES**

35 U.S.C. §102(e) .....	7, 11, 12, 13, 14
-------------------------	-------------------

**I. Real Party in Interest**

The real party in interest for the referenced application is Sun Microsystems, Inc. (“Sun”). An Assignment transferring all interest in the referenced application from the inventors to Sun was filed with the USPTO on November 11, 2001. The Assignment is recorded at Reel 012339, Frame 0073.

**II. Related Appeals and Interferences**

To the best of the knowledge of the Appellant and the Appellant’s legal representative, there are no other appeals or interferences that will directly affect, be affected by, or have a bearing on the decision of the Board of Patent Appeals and Interferences (“the Board”) in this appeal.

**III. Status of Claims**

The present application, Serial No. 09/997,927 (“the ’927 Application”) was filed on November 30, 2001. As filed, the ’927 Application included claims 1–36. Claims 1, 9, 25–30 and 33–36 were cancelled in a response to mailed to the USPTO on May 10, 2005. No additional claims have been subsequently cancelled or amended. Of the remaining claims, claims 8, 24, and 31 are independent.

Claims 2–8, 10–24, and 31–32 are presently pending in the ’927 Application. All of the pending claims, claims 2–8, 10–24, and 31–32, were finally rejected in a final Office Action mailed on August 12, 2005. A Request for a Pre-Appeal Brief was filed with a Notice of Appeal on November 14, 2005. A Notice of Panel Decision from Pre-Appeal Brief was issued December 2, 2005, upholding the final rejections of claims 2–8, 10–24, and 31–32.

**IV. Status of Amendments**

The claims were last amended in a response to the Office Action filed on May 10, 2005. The amended claims were entered and considered in the final Office Action dated August 12, 2005. Therefore, all amendments submitted to the Examiner during prosecution have been entered and are reflected in the Claims Appendix.

**V. Summary of Claimed Subject Matter**

Independent claim 8 relates to a method for transparently injecting proxies into a distributed application, where the distributed application includes a server portion and a client portion. The method includes (i) analyzing the server portion to determine the remote objects present in the server portion and then creating a proxy for each of the remote objects in the server portion, (ii) modifying the client portion such that the calls to the remote objects are substituted (*i.e.*, replaced) by calls to the aforementioned proxies, and (iii) interposing (*i.e.*, placing between) a runtime between the client portion and the server portion, where the runtime includes each of the aforementioned proxies. The method recited in independent claim 8 is discussed in at least paragraphs [0024] - [0026] of the '927 Application.

Dependent claim 10 depends from independent claim 8 and relates to parsing machine code for the server portion during the analysis of the server portion. The method recited in dependent claim 10 is discussed in at least paragraph [0024] of the '927 Application.

Dependent claim 15 depends from independent claim 8 and relates to modifying the client portion to substitute a call to a first lookup service that locates the remote object with a call to a second lookup service that locates the corresponding proxy. Said another way, dependent claim 15 is directed to replacing calls to a first lookup service with calls to a second lookup service, where the first lookup service is configured to search for remote objects and the

second lookup service is configured to search for proxies. The method recited in dependent claim 15 is discussed in at least paragraph [0026] of the '927 Application.

Independent claim 24 relates to a method for transparently injecting proxies into a distributed application, where the distributed application includes a server portion and a client portion. The method includes (i) analyzing the server portion to find each remote object in the server portion, (ii) creating a proxy for each remote object in the server portion and including the proxy in a client runtime library, (iii) analyzing the client portion to determine all the calls made to remote objects in the server portion and replacing the aforementioned calls with calls to the corresponding proxy, and (iv) interposing the client runtime library between the client portion and the server portion. The method recited in independent claim 24 is discussed in at least paragraphs [0024] - [0026] of the '927 Application.

Independent claim 31 relates to a computer-readable medium having recorded thereon instructions executable by a processor. The instructions correspond to instructions for: (i) analyzing a server portion of a distributed application to find each remote object in the server portion, (ii) generating a proxy for each remote object in the server portion (iii) and including the proxy for each remote object in the server portion in a runtime library. The instructions recited in independent claim 31 are discussed in at least paragraphs [0024] - [0026] of the '927 Application.

Dependent claim 32 depends from independent claim 31 and relates to a computer-readable medium having recorded thereon instructions executable by a processor. In dependent claim 32, the instructions correspond to instructions for modifying a client portion of the distributed application such that a call for a remote object is replaced with a call for a corresponding proxy. The instructions recited in dependent claim 32 are discussed in at least paragraph [0026] of the '927 Application.

## VI. Grounds of Rejection to be Reviewed on Appeal

The sole ground of rejection to be reviewed is the rejection of claims 2-8, 10-24, and 31-32 under 35 U.S.C. §102(e) as being anticipated by U.S. Patent No. 6,629,128 (“Glass”).

## VII. Argument

In this appeal, the Applicant argues that claims 2-8, 10-24, and 31-32 are patentable over Glass for the reasons stated below. For the purposes of this appeal, claims 2-8, 11-14, 16-24, and 32 stand or fall together, claim 31 stands or fails on its own, claim 10 stands or falls on its own, and claim 15 stands or falls on its own. Claim 8 is representative of group including claims 2-8, 11-14, 16-24, and 32.

Under 35 U.S.C. §102(e), a claim in a patent application may be rejected if “the invention was described in (1) an application for patent, published under § 122(b), by another filed in the United States before the invention by the applicant for patent....” Furthermore:

Anticipation under 35 U.S.C. § 102 means lack of novelty, and is a question of fact. To anticipate, *every* element and limitation of the claimed invention *must* be found in a *single* prior art reference, arranged as in the claim.

*Brown v. 3M*, 265 F.3d 1349, 1351 (Fed. Cir. 2001) (emphasis added). The Federal Circuit has held repeatedly that anticipation requires disclosure of each and every element of the claimed invention in a single prior art reference. *See, e.g., Schering Corp. v. Geneva Pharms.*, 339 F.3d 1373, 1377 (Fed. Cir. 2003); *Diversitech Corp. v. Century Steps, Inc.*, 850 F.2d 675, 677 (Fed. Cir. 1988); *Orthokinetics, Inc. v. Safety Travel Chairs, Inc.*, 806 F.2d 1565, 1574 (Fed. Cir. 1986).

A. Claims 2-8, 11-14, 16-24, and 32

Claim 8 is representative of this group. Accordingly, the following arguments with respect to claim 8 are similarly applicable to claims 2-7, 11-14, 16-24, and 32.

Independent claim 8 recites, in part: (i) analyzing the server portion to find each remote object in the server portion and creating the proxy object for each remote object in the server portion and (ii) analyzing the client portion to determine calls made to remote objects in the server portion and replacing calls for remote objects with calls for a corresponding proxy object. Each of the aforementioned limitations is addressed below.

(i) “analyzing the server portion to find each remote object in the server portion and creating the proxy for each remote object in the server portion” – This limitation clearly requires that the server portion is analyzed to find each remote object in the server portion to create a corresponding proxy object for each remote object. The Examiner has indicated that this limitation is not equivalent to finding *all* the remote objects in the server portion. (See Office Action mailed August 12, 2005, p. 6). The Applicant asserts that this limitation clearly indicates that *all* remote objects in the server portion are located and corresponding proxy objects are created.

While the Applicant concedes that the term *all* is not listed in the claim limitation, the language of the claim (*i.e.*, analyzing the server portion to find each of the remote objects) read in its entirety is equivalent to find all of the remote objects. Put simply, if “each” of the remote objects have been found, necessarily “all” of the remote objects present in the server portion have been found. Accordingly, the Examiner’s rejection, which is based solely on the lack of the word *all* in the claims, is improper as the Examiner has failed to analyze the claim limitation in its *entirety*.

Further, in view of the Applicant's assertion that the limitation "analyzing the server portion to find each remote object in the server portion and creating the proxy for each remote object in the server portion" requires that "all" of the remote objects present in the server portion have been found, claim 8 is patentable over Glass. Specifically, Glass teaches dynamically creating a proxy for an object as needed at run-time (*See Glass, col. 6, ll. 51-52, and Figure 2*). The method for dynamically creating a proxy, as disclosed in Glass, includes receiving a request for an object, locating the object, determining whether a proxy for the requested object exists, and then either creating a proxy if the proxy doesn't exist or forwarding the request for the object to the correspond proxy if the proxy exists. (*See Glass, Figure 2*).

In contrast, claim 8, as discussed above, requires that the server portion of the distributed application is analyzed to find *each* remote object (as opposed to only remote object objects that are requested at run-time), and that a proxy is created for *each* remote object in the server portion. There is no teaching or suggestion in Glass of analyzing the server portion to find all of the remote objects for the purpose of creating a proxy for all the remote objects. Rather, Glass only teaches searching the client and server to locate a *single* object or to determine the presence of a proxy corresponding to an object. Accordingly, this limitation is not taught or suggested by Glass.

(ii) "analyzing the client portion to determine calls made to remote objects in the server portion and replacing calls for remote objects with calls for a corresponding proxy" – The Applicant respectfully asserts that Glass does not teach replacing calls to remote objects with calls to a proxy. Specifically, the Examiner has asserted that the above limitation is taught by the following statement in Glass:

The system generates and utilizes remote proxy objects in all inter-object communication to provide additional processing support. Thus, any communication between objects, regardless of their location, utilizes remote proxy objects. These remote proxy objects act as a middle man between the requested object and the requesting object to provide additional processing functionality such as increased security. (See Glass, col. 7, ll. 40-47).

The above excerpt indicates generally how proxy objects may be used without any indication that (i) the client portion of the distribution application is analyzed to locate calls to remote objects or (ii) that calls to remote objects located on the client side are replaced with calls to the corresponding proxy objects.

Further, construing the above excerpt of Glass as disclosing the claimed limitations is inconsistent with the specification of Glass. In other words, the Applicant respectfully asserts that the Examiner has misinterpreted the meaning of the above paragraph by ignoring the remaining disclosure of Glass. Specifically, Glass discloses that each time a remote object is requested, the steps in Figure 2 are followed. (See Glass, col. 7, ll. 34-62). One of the steps in Figure 2 is to determine whether a proxy corresponding to the remote object exists (See Glass, Figure 2, Step 30). If Glass actually disclosed analyzing the client portion and replacing calls to the remote object with calls to the proxy object, then Step 30 of Figure 2 would be redundant as there would be no need to check if a corresponding proxy existed because the call to the remote object would already have been replaced with a call to the proxy object. Thus, Glass does not appear to have even contemplated modification to the client portion. Rather, Glass focuses on the dynamic generation of proxies at run-time without any modification to the client portion.

As Glass does not disclose each and every element recited in independent claim 8 of the '972 Application, it is not a proper anticipating reference under 35 U.S.C. §102(e). *See Brown*, 265 F.3d at 1351. Accordingly, Glass is also an improper anticipating reference for claims 2-7, 11-14, and 16-23, which depend, either directly or indirectly, from claim 8. Further, Glass is also an improper anticipating reference for independent claim 24 and dependent claim 32, which depends from independent claim 31. Therefore, Appellant respectfully requests reversal of the rejection of claims 2-7, 11-14, 16-24, and 32 under 35 U.S.C. §102(e).

#### B. Claim 31

Independent claim 31 recites, in part, “analyzing a server portion of a distributed application to find each remote object in the server portion.” This limitation clearly requires that the server portion is analyzed to find each remote object in the server portion to create a corresponding proxy object for each remote object. The Examiner has indicated that this limitation is not equivalent to finding *all* the remote objects in the server portion. (*See* Office Action mailed August 12, 2005, p. 6). The Applicant asserts that this limitation clearly indicates that *all* remote objects in the server portion are located and corresponding proxy objects are created.

While the Applicant concedes that the term *all* is not listed in the claim limitation, the language of the claim (*i.e.*, analyzing the server portion to find each of the remote objects) read in its entirety is equivalent to find all of the remote objects. Put simply, if “each” of the remote objects have been found, necessarily “all” of the remote objects present in the server portion have been found. Accordingly, the Examiner’s rejection, which is based solely on

the lack of the word *all* in the claims, is improper as the Examiner has failed to analyze the claim limitation in its *entirety*.

Further, in view of the Applicant's assertion that the limitation "analyzing the server portion to find each remote object in the server portion and creating the proxy for each remote object in the server portion" requires that "all" of the remote objects present in the server portion have been found, claim 31 is patentable over Glass for at least the same reasons as discussed above with respect to claim 8.

As Glass does not disclose each and every element recited in independent claim 31 of the '972 Application, it is not a proper anticipating reference under 35 U.S.C. §102(e). *See Brown*, 265 F.3d at 1351. Therefore, Appellant respectfully requests reversal of the rejection of claim 31 under 35 U.S.C. §102(e).

### C. Claim 10

Dependent claim 10 recites: "wherein analyzing the server portion comprises parsing machine code for the server portion." The Examiner has asserted that the following portion of Glass teaches this limitation:

The next function invoked by remote proxy generation control module 34 is byte code generator 42. The purpose of byte code generator 42 is to directly generate the executable code corresponding to JClass information 38. Jclass information 38 is the definition of the Java class of which remote proxy object 22 is an instance. That is, Jclass information 38 is the definition of remote proxy class 23. Byte code generator 42 reviews Jclass information 38 and generates the corresponding byte codes, or executable code, into remote proxy class 23 which is equivalent to a Java .class file except that it is not stored on a permanent storage device. (Glass, col. 9, ll. 45-55)

The Applicant respectfully asserts that the above excerpt does not disclose the limitation recited in claim 10. Specifically, the above excerpt teaches the *creation* of byte code by analyzing source code (*i.e.*, JClass Information). The Applicant respectfully notes that byte code is not equivalent to machine code. Further, the above excerpt is completely silent with respect to parsing machine code.

As Glass does not disclose each and every element recited in dependent claim 10 of the '972 Application, it is not a proper anticipating reference under 35 U.S.C. §102(e). *See Brown*, 265 F.3d at 1351. Therefore, Appellant respectfully requests reversal of the rejection of claim 10 under 35 U.S.C. §102(e).

#### D. Claim 15

Dependent claim 15 recites: “modifying the client portion to substitute a call to a first lookup service that locates the remote object with a call to a second lookup service that locates the corresponding proxy.” Thus, claim 15 requires: (i) a first lookup service configured to lookup remote objects, (ii) a second lookup service configured to lookup proxy objects; (iii) the replacement of calls to the first lookup service with calls to the second lookup service. The Applicant respectfully asserts that there is no teaching or suggestion of two lookup services as recited in the claims.

Further, even assuming *arguendo* that Glass discloses two distinct lookup services, there is no teaching of modifying the client portion. In fact, as discussed above, the teachings of Glass are directed to not modifying the client portion.

As Glass does not disclose each and every element recited in dependent claim 15 of the '972 Application, it is not a proper anticipating reference under 35 U.S.C. §102(e). *See*

*Brown*, 265 F.3d at 1351. Therefore, Appellant respectfully requests reversal of the rejection of claim 10 under 35 U.S.C. §102(e).

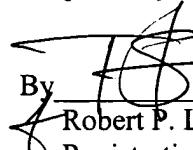
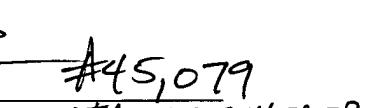
### VIII. Conclusion

For the reasons presented above, claims 2-8, 10-24, and 31-32 of the '927 Application are patentable over the cited art, as Glass does not disclose all of the limitations recited therein. *See Brown*, 265 F.3d at 1351. Therefore, the Appellant respectfully requests that the Board reverse the Examiner's rejections and allow all pending claims 2-8, 10-24, and 31-32 of the '927 Application.

Please apply any charges not covered, or any credits, to Deposit Account 50-0591 (Reference No. 16159/021001).

Date: January 17, 2006

Respectfully submitted,

By   
Robert P. Lord   
Registration No.: 46,479  
OSHA · LIANG LLP  
1221 McKinney St., Suite 2800  
Houston, Texas 77010  
(713) 228-8600  
(713) 228-8778 (Fax)

## Claims Appendix

1. (Cancelled)
2. The method of claim 24, further comprising:  
interposing a server runtime between the client runtime and the server portion that  
enables interaction between the client runtime and the server portion.
3. The method of claim 2, further comprising:  
sending a message to the server runtime to fetch data from the remote object.
4. The method of claim 3, further comprising:  
receiving data fetched by the server runtime and caching data in the proxy.
5. The method of claim 4, further comprising:  
sending a message to the server runtime to synchronize data cached in the proxy with  
data in the remote object.
6. The method of claim 2, further comprising:  
sending a message to the server runtime to invoke a method of the remote object on  
behalf of the proxy.
7. The method of claim 6, further comprising:  
receiving a result of invoking the method of the remote object from the server runtime  
and passing the result to the proxy.

8. A method for transparently injecting proxies into a distributed application having a server portion and a client portion, comprising:  
creating a proxy for each of a plurality of remote objects in the server portion, each proxy implementing an interface of a corresponding remote object and having a capability to cache data from the corresponding remote object;  
modifying the client portion to substitute a call for a remote object with a call for a corresponding proxy; and  
interposing a runtime that includes the proxy for each of the plurality of remote objects between the client portion and the server portion,  
wherein creating the proxy for each of the plurality of remote objects in the server portion comprises analyzing the server portion to determine each of the plurality of the remote objects in the server portion.
9. (Cancelled)
10. The method of claim 8, wherein analyzing the server portion comprises parsing machine code for the server portion.
11. The method of claim 8, wherein analyzing the server portion comprises parsing a descriptor containing a list of classes in the server portion.
12. The method of claim 8, wherein analyzing the server portion comprises parsing source code for the server portion.
13. The method of claim 8, wherein modifying the client portion comprises modifying machine code for the client portion.
14. The method of claim 8, wherein modifying the client portion comprises modifying source code for the client portion.
15. The method of claim 8, further comprising:  
modifying the client portion to substitute a call to a first lookup service that locates the remote object with a call to a second lookup service that locates the corresponding proxy.

16. The method of claim 15, wherein the lookup service that locates the corresponding proxy is included in the runtime.
17. The method of claim 8, further comprising:  
modifying the client portion to substitute a call to manage a lifecycle of the remote object with a call to manage a lifecycle of the corresponding proxy.
18. The method of claim 8, further comprising:  
fetching data from the remote object into the proxy associated with the remote object.
19. The method of claim 18, further comprising:  
synchronizing data in the proxy with data in the remote object associated with the proxy.
20. The method of claim 8, further comprising:  
invoking a method of the remote object on behalf of the proxy associated with the remote object.
21. The method of claim 20, further comprising:  
receiving a result of invoking the method of the remote object and passing the result to the proxy.
22. The method of claim 8, wherein the runtime includes a client runtime that interacts with the client portion and a server runtime that interacts with the server portion.
23. The method of claim 22, wherein the client runtime and server runtime communicate in order to enable interaction between the client portion and the server portion.
24. A method for transparently injecting a proxy into a distributed application having a server portion and a client portion, comprising:  
analyzing the server portion to find each remote object in the server portion;  
creating the proxy for each remote object in the server portion and including the proxy in a client runtime library;  
analyzing the client portion to determine calls made to remote objects in the server portion and replacing calls for remote objects with calls for a corresponding proxy; and

interposing the client runtime library between the client portion and the server portion.

25. (Cancelled)
26. (Cancelled)
27. (Cancelled)
28. (Cancelled)
29. (Cancelled)
30. (Cancelled)
31. A computer-readable medium having recorded thereon instructions executable by a processor, the instructions for:  
analyzing a server portion of a distributed application to find each remote object in the server portion;  
generating a proxy for each remote object in the server portion; and  
including the proxy for each remote object in the server portion in a runtime library.
32. The computer-readable medium of claim 31, further comprising:  
instructions for modifying a client portion of the distributed application such that a call for a remote object is replaced with a call for a corresponding proxy.
33. (Cancelled)
34. (Cancelled)
35. (Cancelled)
36. (Cancelled)

**Related Proceedings Appendix**

Not applicable to the present appeal.